

Lab session Stacks and Subroutines

Group A: November 13, 2009

Group B: November 10, 2009

Work in the given groups of two. Submit your solutions to the respective assignment on Blackboard. The file name is:

`s05_sOXXXXX_sOXXXXX.tar.gz`

One of the group members commits your solution. Keep an eye on the deadline (see Blackboard)!

1 Exercises

1. Write a MIPS program that implements a simple program that prints out the English word for an entered number. If the user enters “2”, the output is “two”. The program gives output for each number from 0 to 4.
 - Use if-then-else operations to implement this.
 - Use a case table to implement this. Hint: use `la` and `jr`.

Make a comparison of these two solutions in terms of (a) number of instructions used (size of the program) and (b) number of instructions executed (performance of the program). Compare in both cases the worst case and best case scenarios (with respect to the input value) of the two implementations. Give your conclusion.

2. Write a MIPS program that implements and uses a simple stack of integers. Simply provide the following instructions:

```
push 7
push 15
pop
push 31
pop
push 63
pop
pop
```

3. Implement a leaf procedure.

- Implement a procedure that prints a given string backwards to the console. The argument is the string, and there is no return value. Make sure that the machine state (i.e. the registers) must be remembered on the stack in order to acquire the storage space in the registers that you need for this procedure. Afterwards, the machine state and stack should be restored.

Note that the argument can be arbitrarily long, so the standard \$a0 to \$a3 registers won't suffice to pass the arguments. Think of a way to solve this problem.

- Call the procedure.

2 Project

There is no project this week. You only have to submit your solutions to the exercises. There will be no feedback loop on this lab session.